

## REMARKS

This Amendment is in response to the Office Action dated October 18, 2005. Claims 1-20 are pending. Claims 1-20 are rejected. Accordingly, claims 1-20 remain pending in the present application.

### Present Invention

Hereinbelow, Applicant will describe with particularity why Nielsen neither teaches nor suggests the invention as recited in the claims.

### Regarding independent claims 1, 9 and 15:

- “forming a database of font specifications”

Nielsen does not teach or suggest forming a database of font specifications as recited in the claims; rather his application teaches forming a database of font size viewing preferences. The viewing preferences that Nielsen records per URL have nothing to do with accurately reproducing the document via rendering to the computer screen or printed page as it was designed. The font size preference Nielsen records are not related in any way to the fonts contained in the content as it was designed to be viewed, and are only recorded IF a user makes a change to viewing size of a web page at run time. In fact the document content could be altered 100% and this would not impact the stored font size preferences as they are associated exclusively with a network address (URL) and there is no way in Nielsen to determine if the content associated with the URL had changed. Also, the font sizes Nielsen stores are viewing preferences not “font specifications”. Font specifications are the metadata derived from the font employed in originally generating the document.

- “accessing the database when opening documents to ensure usage of proper fonts”

Nielsen does teach accessing a database to apply font size preferences to whatever is rendered but does not teach ensuring proper usage of fonts. Nielsen does not teach or suggest “accessing the database to ensure usage of proper fonts” which allows for retrieving from a database via multiple font specifications an unambiguous lookup of the exact font(s) used to create the document, activating that/those font(s) into computer memory so that the font(s) that was/were originally used to create the document can be used to render it again so it appears exactly the way the designer intended it to be viewed. Nielsen teaches associating a single font size specification with a URL so that whatever is rendered at a particular web address is shown in a certain size.

- “accessing the database when saving documents to ensure usage of proper fonts”

Nielsen teaches recording a viewing preference change per URL if one is made by the user. Claims 1, 9 and 15 recite “accessing the database when saving documents to ensure usage of proper fonts” which allows for the recording an unambiguous font identifier comprised of font specifications for each font used in composing the document in the document as it is saved so that we can quickly and efficiently ensure that we can either reproduce the document exactly the same or can tell a user exactly what fonts are missing. Nielsen does not teach or suggest this.

### Summary

- Nielsen does not teach accurate font matching; i.e. finding and activating in computer memory the exact fonts that went into producing a document/some content.
- Nielsen does not teach decomposing a typeface into multiple font specifications and storing those in a database.

- Nielsen does not teach assembling an unambiguous identifier from those font specifications to exactly match fonts to ensure proper usage.
- Nielsen does teach accessing a database, but only to employ a font size preference over ride as determined by the viewer of the document/content (not the creator or modifier/editor of the document as we teach) for rendering a document ensuring proper font usage.

Regarding dependent claims 2, 10 and 16:

Nowhere in Nielsen does it teach or suggest obtaining a font list for a document, storing a font list for a document, or determining whether font specifications for each font in the font list exist in the database as recited in the claims. Figures 3A and 3B of Nielsen clearly and unambiguously indicate that Nielsen does not store any specific font information related to what is being viewed. The data recorded are the URL, a font size viewing preference and a date that the change was made. There are no font specifications (postscript name, family name, foundry name, version, kerning checksum, outline file size, font type, etc.), recorded in Nielsen. According to the description it would be impossible to generate a font list from the recorded information, further the font size viewing preference would have to be universally applied to all fonts on the page.

Nielsen teaches storing a viewer specified font size preference associated with a URL; the information being stored by Nielsen is unrelated to what our claim covers. In the recited invention, a list of font identifiers are stored in a document that enumerate the exact fonts that went into creating the document. When the document is opened that list of font identifiers is looked up in the local font database and all of them that can be found are activated and the user is warned if the fonts necessary to render the document as it was originally designed are not found.

Summary

Nielsen does teach accessing a database with data in it, hardly a novel claim. Nielsen also teaches reading and writing information to the database, this information is not related to specific fonts but rather viewing preferences for a web browser. Nielsen does not teach obtaining a font list for a document, extracting the meta data embedded in the font file, or uniquely and unambiguously identifying each font in a document.

Regarding claims 3, 11 and 17:

Nielsen does not store information about every single font embedded in a document (Figures 3A and 3B). Nielsen stores information about a URL viewed and records a font size preference only if it is changed by the viewer. There is no way a user of Nielsen's invention could retrieve font specifications for a font as the preference setting is globally tied to a URL not to a specific font as recited in claims 3, 11 and 17.

Summary

Nielsen doesn't store font specifications. The data stored in Nielsen has nothing to do with the actual font files, Nielsen stores viewing preferences associated with URLs. Nielsen does not teach retrieving font specifications for each font or fonts themselves. The text in diagram element 430 in figure 4 is misleading- it indicates that the file is displayed by using the font stored in the database. However, referring to the paragraph in column 5, lines 17-34, describing the adaptive user interface, it clearly indicates that "If a subsite preference is found (this would be the URL/font size key value pair) then the requested page is displayed using the preferred font size." Nielsen clearly does not teach retrieving each the font specifications for each font from the database.

Regarding claims 4, 12 and 18:

Nielsen does not create font specifications for each font. Nielsen creates a font size viewing preference, unrelated to any specific fonts being viewed, for a URL entry (Figures 3A – 3B) only if the default has been changed for the page.

In a method in accordance with the present invention, the font specifications (Postscript name, foundry, kind [PostScript, TrueType, etc], glyph data, horizontal metrics, vertical metrics, encoding, kerning, etc.) are created for each font typeface by extracting this information from a font file either in memory or on disk that was used to create or edit the document; parse those specifications and combine them into a unique font identifier, and write that unique font identifier into the document. The extracted metadata for the fonts used in creating the document are stored in a database. Sometimes there are multiple typefaces per physical font file (a font suitcase), where we will have to extract the font specifications for all typefaces in the file and partition the typefaces into separate files so that we can activate only the typefaces necessary to render the document.

Summary

Nielsen does teach writing a URL/font size viewing preference into a database. As discussed earlier the font size viewing preference has nothing to do with an individual font, but rather a preference size associated with a URL (a web page) that could have multiple fonts listed on it. Claims 4, 12 and 18 recites creating the font specifications for each font when the font specifications do not exist and storing those elements in addition to the Postscript name so that we can uniquely and unambiguously identify a font.

Regarding claims 5, 13 and 19:

The examiner claims that Nielsen discloses writing the font information into the document, this is not so. Nielsen clearly indicates that the font size viewing preference (only if changed) is written into the URL/preference database the examiner cites (figure 5C, #542-#544) which show the preferences being written into the database.

Claims 5, 13 and 19 recites writing a font specification into the document. This allows for writing the font metadata (the font specifications) into a font database so that when a document is exchanged from one computer to another the document font list embedded inside the document can be compared with the local font database external to the document to determine whether or not the exact same fonts are on the viewing computer so that the viewer and the creator can view and print the document and it will be exactly the same.

Summary

Nielsen does not teach or suggest writing the font specification into the document. Nielsen teaches writing URL/font size viewing preference into a database. Claim 5 recites writing the font specification into the document and saving the document. For example, this can be accomplished by taking a checksum of the individual key elements that make up a font that will uniquely identify it and writing those elements into a document file for each font in the document in the form of a slug. That slug can be retrieved when a document is opened via an auto-activation plug-in and then ensure that the fonts in question are loaded into system memory so the document can be displayed in the exact format / layout that it was created in.

Regarding claims 6, 14 and 20:

Opening a document and retrieving the unique font identifiers for all fonts that compose the document, checking the local font database to determine the exact set of fonts that need to be activated in computer memory, and rendering the document as recited in the claims - is not the same or even related to changing a font size afterwards (as Nielsen teaches). Further Nielsen only allows for storage of one font size viewing preference per URL. In order to extract font information from a document and ensure that the those fonts necessary to display the document are active in computer memory prior to rendering the document requires the word processing application either directly communicates with the font database or that we have a process running inside the word processing applications process space (a plug-in) that can block rendering a page/the document until we have the correct font(s) loaded.

#### Summary

Nielsen teaches that when a URL is hit via a web browser that a local database is queried to determine whether or not the URL has been hit before and whether or not there is a font size preference for viewing. Nielsen does not teach opening a document and analyzing all of the fonts embedded in the document and ensuring that those fonts are active in memory (if they are available for activation). Nielsen further does not teach submitting font identifiers to a database, identifying whether or not those fonts can be located, and downloading fonts to the client machine, activating those fonts on the operating system.

#### Regarding claim 7:

The examiner states that Nielsen discloses searching the database to locate each font specified by font specifications in the document. How? Nielsen doesn't store font specific information at all. Figures 3A and 3B clearly indicate that font specific information is not stored

per document. Even if the examiner generalizes that a document = URL, only a font size is stored in the database – and this is only a viewing preference that would exist on the local machine if a viewer decided to change the size. Nielsen describes an adaptive user interface feature. Nielsen can't find a single font in any document on any URL ever because the information is not being recorded.

### Summary

When a document is opened on a computer PC/Mac the application scans the document for fonts in the document and scans memory for active fonts; if it discovers that fonts are necessary in order to render the document it will send messages to the operating system requesting those fonts or it will replace the fonts that are missing in the document with the closest matches to the ones active in the operating system. In the present invention these unique font identifiers are written into the database for each font, either by using an ambiguous matching methodology (the postscript name) or an unambiguous methodology (using the unique identifier per font embedded in each document as a slug).

### Regarding claim 8:

Nielsen doesn't teach or suggest retrieving fonts, only viewing preferences data from a preferences database (which only exists if the viewer changed the preference for that URL previously). Retrieving a font involves taking the unique font identifier in the document, comparing it to the font database to determine whether or not there is a match, if there is a match, then taking the font and activating the font in computer memory so it is available to the word processor to render the document accurately. If the font is not available, then the viewer is presented with the best available match for a substitution.



Summary

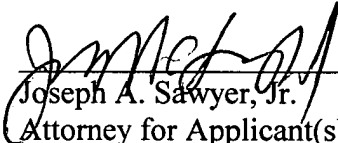
Nielsen does not teach or suggest retrieving a font, merely retrieving a viewing preference (font size). That viewing preference does absolutely no good in displaying a web page if the font necessary to render that web page is not active in the operating system first. In the present invention the font is identified and the font file is retrieved which is then plugged into the operating system via activation making it available to all applications on the computer that need to render a document (or web page) with that font embedded in it.

In view of the foregoing, it is submitted that the claims 1-20 are allowable over the cited references and are in condition for allowance. Applicant respectfully requests reconsideration of the rejections and objections to the claims, as now presented.

Applicants' attorney believes this application in condition for allowance. Should any unresolved issues remain, Examiner is invited to call Applicants' attorney at the telephone number indicated below.

Respectfully submitted,  
SAWYER LAW GROUP LLP

March 14, 2006

  
\_\_\_\_\_  
Joseph A. Sawyer, Jr.  
Attorney for Applicant(s)  
Reg. No. 30,801  
(650) 493-4540